

# Sound Creator

# Sound Editor

Gültig ab Software Version  
ZSP V1.26.00

AUSGABEN:

2022 Oktober  
2024 März  
2025 Juni  
2026 April

# Scripts

# BETRIEBSANLEITUNG

Einführung in die Programmierung  
von Scripts  
in ZIMO Sound Creator (ZSP)  
und ZIMO Sound Editor (ZPP Konfig)

Statenname	Statenummer	Kommando	Parameter
Thyristor Sample	0	Silence.wav	mit Loop starten
Ahfrage Parameter	1	HLU-Limit	ist gleich Fixwert Halt (HLU, ABC, DC) dann gehe zu State 0
Ahfrage Parameter	2	Ziergeschwindigkeit	ist gleich Fixwert 0 dann gehe zu State 1
Thyristor Sample	3	Taurus_Thyristor_01 fade-in_02.wav	mit Loop starten
Ahfrage Parameter	4	Fahrgeschwindigkeit	ist größer als Fixwert 12 dann gehe zu State 6
Gehe zu State	5	4	
Thyristor Sample	6	Taurus_Thyristor_02_04.wav	mit Loop starten
Ahfrage Parameter	7	Fahrgeschwindigkeit	ist größer als Fixwert
Ahfrage Parameter	8	Fahrgeschwindigkeit	ist größer als Fixwert
Ahfrage Parameter	9	Fahrgeschwindigkeit	ist ungleich Fixwert
Gehe zu State	10	0	
Thyristor Sample	11	Taurus_Thyristor_03_06.wav	mit Loop starten
Ahfrage Parameter	12	Fahrgeschwindigkeit	ist größer als Fixwert
Ahfrage Parameter	13	Fahrgeschwindigkeit	ist größer als Fixwert
Gehe zu State	14	6	
Thyristor Sample	15	Taurus_Thyristor_04_08.wav	mit Loop starten
Ahfrage Parameter	16	Fahrgeschwindigkeit	ist größer als Fixwert
Ahfrage Parameter	17	Fahrgeschwindigkeit	ist größer als Fixwert
Gehe zu State	18	11	
Thyristor Sample	19	Taurus_Thyristor_05_10.wav	mit Loop starten
Ahfrage Parameter	20	Fahrgeschwindigkeit	ist größer als Fixwert
Ahfrage Parameter	21	Fahrgeschwindigkeit	ist größer als Fixwert
Gehe zu State	22	15	
Thyristor Sample	23	Taurus_Thyristor_06_12.wav	mit Loop starten
Ahfrage Parameter	24	Fahrgeschwindigkeit	ist größer als Fixwert
Ahfrage Parameter	25	Fahrgeschwindigkeit	ist größer als Fixwert
Gehe zu State	26	19	
Thyristor Sample	27	Taurus_Thyristor_07_14.wav	mit Loop starten
Ahfrage Parameter	28	Fahrgeschwindigkeit	ist größer als Fixwert
Ahfrage Parameter	29	Fahrgeschwindigkeit	ist größer als Fixwert
Gehe zu State	30	36	

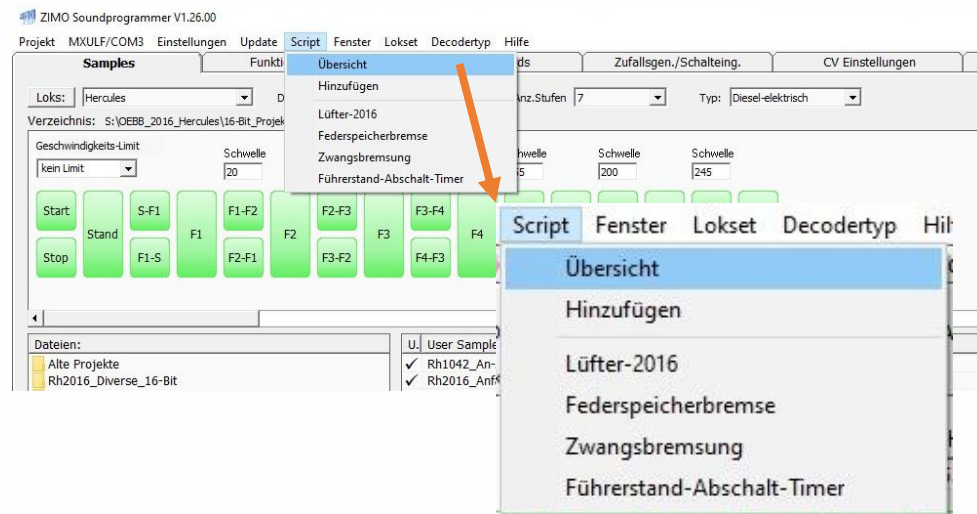
Derzeit (Stand: April 2026) trägt der **ZIMO Sound Creator** noch den Namen **ZSP** und der **ZIMO Sound Editor** noch den Namen **ZPP Konfig**. Die Programme sind als Bestandteil des Softwarepakets gemeinsam auf der Website [www.zimo.at](http://www.zimo.at) erhältlich. Künftig werden sie **getrennt verfügbar** sein und unter ihren **neuen Bezeichnungen** veröffentlicht. Dies wird keinen Einfluss auf die Funktionalität der Scripts haben, da diese in beiden Programmen gleich erstellt werden.

## Allgemeines:

Mit dem **ZIMO Sound Creator (ZSP)** und den **ZIMO Sound Editor (ZPP Konfig)** kann man mit Hilfe von **Scripts** vorprogrammierte Abläufe und Abhängigkeiten erstellen und damit die Funktionen der ZIMO Decoder deutlich erweitern. Die beiden Programme bieten die Möglichkeit, die gewünschten Abläufe in einer Eingabemaske zu erstellen. Je Soundprojekt oder CV-Set ist es bei MS/FS-Decodern ab MS-FW 4.207 möglich, bis zu 16 Scripts zu erstellen (oder zu importieren) und bei MX-Sound-Decodern ab MX-FW 39.0 bis zu 8 Scripts. MX-Sound-Decoder unterliegen jedoch einigen Einschränkungen im Vergleich zur MS-Decoder-Familie; z.B. stehen nur 6 unabhängig abspielbare Soundkanäle zur Verfügung und es können in einem Script nur in bestimmten Fällen zwei Samples gleichzeitig abgespielt werden. Für beide Decoder-Familien gilt: pro Script kann man bis zu **256 States** (Befehle bzw. Zeilen) programmieren. Deaktivieren kann man die einzelnen Scripts über die zugehörigen Bits von CVs **#837** und **#843** (CV #837 Bit0-7 = Scripts 1-8, CV #843 Bit0-7 = Scripts 9-16).

Viele Werte und Parameter können als **Script-CV (#980 bis #1019)** ausgeführt werden, welche man später im Decoder ändern kann, ohne das gesamte Soundprojekt neu laden zu müssen. Häufige Anwendungen für die Verwendung von Script-CVs sind Lautstärken von Samples, Timerlängen oder Fahrstufen. Natürlich müssen aber die erlaubten Werte der einzelnen CVs für die sie stehen beachtet werden - siehe dazu die Decoderanleitungen für [MX-](#) und [MS-Decoder](#).

Ab der MS-FW 4.250 können auch Scripts in **CV-Sets** in **MN-Decoder** verwendet werden. CV-Sets mit Scripts können nur im **ZIMO Sound Editor** erstellt und über das **KLUG** direkt in den Decoder geladen werden. Eine andere Möglichkeit ist die Verwendung des **CV-Set Creators**, mit dem man eine spezielle .zpp-Datei erstellt und das CV-Set mittels USB-Stick über MXULF auf den MN-Decoder lädt. Allerdings ist hier zu beachten, dass bei MN-Decodern keine Scripts mit Sounds möglich sind, bzw. die entsprechenden Befehle (Abfrage nach dem Fahrsound, nach Dieselstufen, Thyristor- und Motor-Samples, etc.) ignoriert werden. Es macht eher Sinn, Funktionsausgänge oder Fahreigenschaften von MN-Decodern über Scripts zu beeinflussen. Man kann sich aber in Einzelfällen mit Timern in der Länge von Sound-Samples behelfen.



## Scripts erstellen und bearbeiten:

Die Erstellung und Bearbeitung von Scripts ist ab Version V1.21.22 sowohl in ZIMO Sound Creator als auch in ZIMO Sound Editor möglich. Im ZIMO Sound Editor können bestehende Scripts in Coded-Projekten allerdings nicht eingesehen, bearbeitet oder gelöscht werden, da diese vom Sound-Provider speziell an den Ablauf des Projekts angepasst sind. Es ist lediglich möglich, sie über die entsprechenden Bits in den CVs **#837** und **#843** zu deaktivieren, was jedoch die Funktionalität des Soundprojekts erheblich einschränken kann. Eigene Scripts können aber auch in Coded-Projekten erstellt oder importiert werden.

Die Scripts findet man im **ZIMO Sound Creator** im Menüpunkt **Script**. Dort wählt man entweder die **Übersicht**, oder **Hinzufügen**, oder wählt direkt eines der vorhandenen Scripts zum Bearbeiten aus. Mit dem Klick auf Übersicht öffnet sich das separate Script-Hauptfenster, welches eine Übersicht bietet.

Im **ZIMO Sound Editor** findet man die Scripts unter dem Register **Scripte**. Die hier beschriebenen Bearbeitungsmethoden sind in beiden Programmen gleich.

Um ein neues Script zu erstellen, auf die Schaltfläche **Hinzufügen** klicken und es erscheint ein Feld mit „Script1“ (oder der nächsten freien Script-Nummer). Wenn ein Script in der Übersicht angewählt ist (blau unterlegt) kann man über Taste F2 dem Script einen charakteristischen Namen geben, damit man es in weiterer Folge leichter erkennt.

Mit Doppelklick auf ein Script öffnet sich der Script-Editor.

Aktiv	Name	Tasten	Ausgänge	CVs	Sounds
<input checked="" type="checkbox"/>	Lüfter-2016	F12		981,982	Rh2016_Lüfter_Stand_02.wav, Rh2016_Lüfter_F0-F2_Start_10.wav, Rh2016_Lüfter_F0-F2_Stop_10.wav
<input checked="" type="checkbox"/>	Federspeicherbremse	F19		988,989,990	Rh2016_Federspeicherbremse_01.wav, Rh2016_Federspeicher-anlegen_01.wav
<input checked="" type="checkbox"/>	Zwangsbremmung	F20		983,984	Rh2016_SiFa-Zwangsbremmung_01.wav, Rh2016_Störung_01.wav, Rh2016_SiFa-Zwangsbremmung_02.wav
<input checked="" type="checkbox"/>	Führerstand-Abschalt-Timer			991	

**Exportieren:** Ein Script kann an einen beliebigen Speicherort exportiert werden, um es beispielsweise in einem anderen Soundprojekt oder dem CV-Set Creator zu verwenden oder mit einer anderen Person zu teilen. Es wird im .zsc-Dateiformat mit beliebigem Dateinamen gespeichert.

**Importieren:** Wenn ein Script gespeichert oder exportiert wurde, kann man es in ein anderes Soundprojekt oder CV-Set importieren. **ACHTUNG:** allgemein wird zu Sound-Samples nur ihre Index-Nummer innerhalb des aktuellen Soundprojektes abgespeichert und diese kann von Projekt zu Projekt unterschiedlich sein. Daher müssen die Sound-Samples nach dem Import in ein anderes Projekt manuell überprüft und gegebenenfalls angepasst werden!

**Fenstergröße:** Das Fenster des Script-Editors kann man in der Größe verstellen und auch maximieren. Dies bietet bei langen und komplexen Scripts den Vorteil, dass mehr States auf einen Blick sichtbar sind.

**Löschen:** Ein Script kann durch einfaches Anklicken angewählt (blau unterlegt) und anschließend mit der Entf-Taste gelöscht werden.

In der Übersicht werden zusätzlich alle mit dem jeweiligen Script verknüpften Elemente angezeigt. Dazu gehören Funktionstasten, Funktionsausgänge, verwendete Script-CVs sowie zugeordnete Sound-Samples.

**Aktivieren / Deaktivieren:** In der Übersicht gibt es die Möglichkeit ein Script über das Häkchen vorne zu aktivieren oder zu deaktivieren. Dies wirkt sich auch direkt auf die Werte der CVs **#837** und **#843** aus.

### Script Editor



#### States hinzufügen oder löschen:

Durch Klick auf einen neuen leeren State, öffnet sich das Auswahl-Menü für die diversen Befehle:

Statenname	Statenummer	Kommando	Parameter
0	Sound	Abfragen	Setzen
0	Sound starten	Abfrage Taste	Lok steuern
0	Sound beenden	Abfrage Parameter	Timer laden
0	Auf Sound Ende warten	Abfrage Reed-Input	Diesel-Stufe fixieren
0	Thyristor Sample	Abfrage Event	Ausgang einschalten
0	Auf Thyristor Ende warten	Abfrage Variable	Event setzen
0	Thyristor Tonhöhe	CV lesen (nur MS)	FA Effekt setzen
0	Bremsenquietschen unterdrücken	CV-Bit lesen (nur MS)	Variable setzen
0	EMotor Sample		Variable/Event ändern
0	Auf EMotor Ende warten		Set Parameter
0	Diesel-Ablauf übernehmen		Set Pwm Gruppe
0	Set Lautstärke (nur MS)		CV schreiben (nur MS)
			CV-Bit schreiben (nur MS)
			State löschen

Durch Klick mittels **linker Maustaste** auf die Nummer eines vorhandenen States, kann man oberhalb von diesem einen neuen State einfügen oder mit Klick auf einen leeren State diesen löschen.



Mit Klick mittels **rechter Maustaste** auf die Nummer eines States gibt es mehr Auswahlmöglichkeiten.

Links unten im Script-Editor befindet sich das Feld „Simulation“. Dieses wird später näher erläutert, da es dazu dient, ein fertiggestelltes Script testweise auszuführen.

Daneben gibt es die Möglichkeit, Schriftart und -größe des Textes nach eigenen Vorlieben anzupassen.

### Scripts schreiben und lesen:

Scripts „liest“ man am besten wie einen Text: man beginnt links oben und liest es Zeile für Zeile, bis rechts unten. Genauso machen es auch die ZIMO Programme, und später der Decoder, beim Abspielen der Scripts. Bei jedem **Befehl** können je nach Art mehrere **Parameter** abgefragt werden. Prinzipiell kann man mit den Scripts sehr viele verschiedene Abläufe generieren. Bei der Konfiguration gibt es allerdings ein paar Grundregeln, die man beachten sollte.

#### Das Gegenteil abfragen

Scripts sind ähnlich dem Prinzip if/else in der Programmierung. Als Beispiel: wenn das Einschalten einer Taste etwas auslösen soll, sollte bis zum Einschalten der Aus-Zustand der Taste abgefragt werden.

an dieser Stelle fort. Wenn die Abfrage allerdings negativ ist, wird der Ablauf im nächsten darunter liegenden State fortgeführt. Im folgenden Beispiel bleibt der Ablauf im gleichen State bis Taste F2 eingeschaltet wird:



Der Befehl „**Abfrage Taste**“ überprüft, ob die Funktionstaste **F2** **ausgeschaltet** ist. Ist dies der Fall, springt das Script zu **State 0**, also bleibt in der aktuellen Zeile.

Wird **F2** nun – entweder am Fahrpult der Digitalzentrale oder über das Online-Fahrpult vom ZIMO Sound Creator oder ZIMO Sound Editor – **eingeschaltet**, fällt die Abfrage negativ aus und das Script läuft automatisch zur nächsten Zeile, also zum nächsten State.

Im nächsten State sollte dann eine beliebige Aktion definiert sein, die bei aktivierter Funktionstaste ausgeführt werden soll – beispielsweise der Start eines Sounds oder ein weiterer Ablauf.

Nach erfolgter Aktion muss der Zustand der Taste erneut abgefragt werden. Im gezeigten Beispiel geschieht dies mit dem Unterschied, dass der Zustand nun mit **EIN** geprüft werden muss:

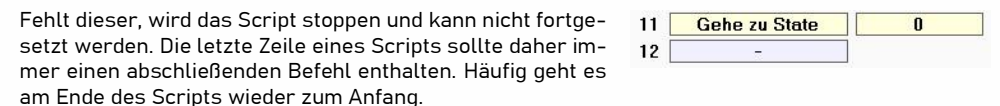


#### Was in einem Script gestartet wird, muss auch dort beendet werden!

Egal ob es sich um Sound-Samples handelt, Funktionsausgänge, das Fahrverhalten, oder den Eingriff in den Ablauf: wenn eine Aktion in einem Script ausgelöst wird, muss man sicherstellen, dass sie auch wieder dort beendet wird.

#### Das Ende eines Scripts darf nicht offen bleiben!

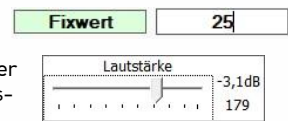
Ein abschließender Befehl (z. B. ‚Gehe zu State X‘) ist zwingend erforderlich, um einen kontinuierlichen Ablauf zu gewährleisten.



Fehlt dieser, wird das Script stoppen und kann nicht fortgesetzt werden. Die letzte Zeile eines Scripts sollte daher immer einen abschließenden Befehl enthalten. Häufig geht es am Ende des Scripts wieder zum Anfang.

Es ist aber nicht zwingend erforderlich, dass vom letzten State aus zu State 0 gesprungen wird; entscheidend ist, dass das Script einen geschlossenen Kreislauf bildet und an keiner Stelle offen bleibt oder stoppt.

Parameter, die man als Fixwert selbst in das Script schreibt, können über Klick in das jeweilige Feld geschrieben und geändert werden.



Lautstärken werden entweder als Fixwert über den Schieberegler, oder über die Auswahl einer Script-CV eingestellt. Wenn eine Script-CV ausgewählt ist, kann die Lautstärke auch nachträglich geändert werden, wenn das Projekt schon auf den Decoder geladen wurde. Dies gilt für alle Parameter, die als CV #980-#1019 ausgewählt werden können und dies wird auch von ZIMO empfohlen. Nachträgliche Änderungen können viel leichter am Decoder durchgeführt werden, ohne dass man komplette Projekte neu laden muss.

## Beschreibung der Befehle:

### Sound:

**Sound starten:** Startet einen **Sound**, der im Parameter „Samples“ ausgewählt wird, in dem alle Samples des Soundprojekts aufgelistet sind. Solange kein Sound ausgewählt wurde, wird das Feld in Rot **Bitte Sample wählen** angezeigt. Der nächste Parameter bestimmt die **Lautstärke**, mit der der Sound abgespielt wird. Wie üblich kann hier nur leiser gestellt werden. Durch Klick in das Lautstärke-Feld kann man entweder über den Schieberegler die Lautstärke reduzieren, oder eine der CVs #980-#1019 auswählen, über deren Wert die Lautstärke verringert wird. Anschließend kann eingestellt werden, ob der Sound einmal oder mit Loop abgespielt werden, oder wie oft dieser wiederholt werden soll. Bei **„Sound einmal abspielen“** wird der Sound ohne weitere Befehle einmal komplett von Anfang bis Ende abgespielt. Der nächste Parameter **„Bei Ende...“** hat hier keinen Einfluss. Ist ein Sound mit Loop-Markern ausgestattet und wird dieser mit **„Loop bis Sound Beenden“** gestartet, gibt es zwei Möglichkeiten ihn zu beenden: **„Bei Ende alles abspielen“** lässt den laufenden Loopbereich in voller Länge zu Ende abspielen, **„Bei Ende vorspringen bis Loop-Ende“** beendet den laufenden Loop vorzeitig und springt zum Endmarker im Sample. Wenn ein Sound-Sample hinter einem Endmarker noch weitergeht, wird dieses Ende ebenfalls noch abgespielt. Die Loop-Marker der Sound-Samples müssen wie gewohnt in ZSP bearbeitet sein, oder beim Hinzufügen von eigenen Samples im ZIMO Sound Editor eingerichtet werden. Sounds oder Loops können auch mit einer fixen **„Loop-Anzahl“** (2-8) gestartet werden. Hierbei wird entweder das komplette Sound-Sample x-mal wiederholt, oder wenn vorhanden der Loopbereich. Als letzten Parameter kann man bestimmen, ob der **„Sound sofort starten“** soll, oder das Ende des zuvor gestarteten Sounds abgewartet werden soll. Mit **„Sound starten sobald letzter Sound beendet ist“** kann im Script die Befehlszeile „auf Sound Ende warten“ eingespart werden.

**Sound beenden:** Wenn der Sound mit Loop gestartet wurde, muss dieser Befehl gesetzt werden, um den Sound wieder zu beenden. Wenn nur einmalig oder mit einer bestimmten Anzahl an Wiederholungen gestartet wurde, ist dieser Befehl nicht notwendig; der Sound wird automatisch beendet.

**Auf Sound Ende warten:** Bevor ein neues Sound-Sample gestartet wird, kann hiermit so lange gewartet werden, bis das aktuelle Sound-Sample (einschließlich aller eingestellten Loops und Wiederholungen) vollständig abgespielt wurde. Zudem sorgt der Befehl für eine Audio-Transition zum nächsten Sound-Sample – Soundlücken werden so vermieden. Für den Ablauf des Scripts hat dieser Befehl jedoch keine Relevanz; die dann folgenden Abfragen laufen weiter!

**Thyristor Sample:** Wählt ein Sound-Sample für den Thyristoreffekt aus. Die Tonhöhe kann entweder über die entsprechenden CVs #289-292 oder über den Script-Befehl „Thyristor Tonhöhe“ gesteuert werden. Die dynamische Lautstärke wird über CVs #293-295, #357-358 eingestellt. Das Programm unterscheidet nicht, welches Soundprojekt aktuell bearbeitet wird, weshalb diese Funktion auch bei Dampf- und Dieselpunkten sichtbar ist. In der Regel werden Thyristor-Samples **„mit Loop gestartet“**. Sollten manuelle Übergänge, beispielsweise zwischen einzelnen Thyristor-Stufen, gewünscht sein, kann auch **„Sound einmal abspielen“** aktiviert werden. Jedes im Script neu gestartete Thyristor-Sample ersetzt das vorherige (auch jenes im Ablauf), sodass kein manuelles Beenden eines laufenden Samples erforderlich ist.

**Auf Thyristor Ende warten:** Wenn ein Thyristor-Sample mit „Sound einmal abspielen“ gestartet wurde, kann mit diesem Befehl der Ablauf so gesteuert werden, dass das vollständige Abspielen dieses Samples abgewartet wird, bevor das nächste Thyristor-Sample gestartet wird. Ohne diese

Sound
Sound starten
Sound beenden
Auf Sound Ende warten
Thyristor Sample
Auf Thyristor Ende warten
Thyristor Tonhöhe
Bremsenquietschen unterdrücken
EMotor Sample
Auf EMotor Ende warten
Diesel-Ablauf übernehmen
Set Lautstärke (nur MS)

Wartefunktion würde das laufende Sample sofort durch das neu gestartete ersetzt, was entweder zu einem vorzeitigen Abbruch oder im ungünstigsten Fall zu einer hörbaren Soundlücke führen kann. Thyristor-Samples, die mit Loop gestartet wurden, können so aber nicht abgewartet werden.

**Thyristor Tonhöhe:** Mit diesem Befehl beeinflusst man den Anstieg der Tonhöhe des Thyristors, gebunden an bestimmte Fahrstufen. So definiert man in Parameter 1 die Tonhöhe (0-255 entsprechen der Tonhöhe in %) des Thyristors bei mittlerer Fahrstufe und in Parameter 2 die maximale Tonhöhe des Thyristors bei maximaler Fahrstufe. Die mittlere Fahrstufe wird in Parameter 3 definiert (0-255 entsprechen der internen Fahrstufe), die maximale Fahrstufe in Parameter 4. Die Einstellungen hier entsprechen den CVs #291-293 und #838. Einmal in einem Script verwendet, kann die Wirkung dieser CVs nicht mehr an den Ablauf des Decoders zurückgegeben werden und alle weiteren Änderungen müssen erneut im Script verwirklicht werden.

**Bremsenquietschen unterdrücken:** Legt fest, ob das Bremsenquietschen im Ablauf beim Verzögern unterdrückt werden soll.

**EMotor Sample:** Hiermit wählt man ein Sound-Sample für den E-Motor aus und startet es wie Thyristor-Samples entweder **„mit Loop“** oder lässt es **„einmal abspielen“**. Die dynamischen Einstellungen für den E-Motor werden über die entsprechenden CVs (#296-299, #372, #373, #844) gesteuert. Wie bei den Thyristor-Samples ersetzt jedes neu gestartete E-Motor-Sample das vorherige.

**Auf E-Motor-Ende warten:** Wenn ein E-Motor-Sample mit „einmal abspielen“ gestartet wurde, wird mit diesem Befehl gewartet, bis es komplett zu Ende abgespielt wurde.

**Diesel-Ablauf übernehmen:** Sobald der Ablauf mit „Automatik deaktivieren und Diesel-Sounds manuell im Script starten“ übernommen wird, wird der Audio-Kanal des Ablaufs (Diesel-Motor oder auch Lüfter von Elektro-Projekten) an das Script übergeben. Noch laufende Ablauf-Geräusche müssen mit „Sound beenden“ gestoppt und der gewünschte neue Ablauf anschließend über Sound-Samples nachgebildet werden. Mit „Diesel-Ablauf wieder an Automatik übergeben bei Stufe“ wird an den Ablauf des Decoders zurück gegeben. Hier kann noch eine Diesel-Stufe definiert werden, in welcher der Ablauf anknüpfen soll. Um sicherzustellen, dass an die korrekte Diesel-Stufe übergeben wird, kann vorab der Befehl „Diesel-Stufe fixieren“ verwendet werden.

**Set Lautstärke** (nur bei MS/FS): Setzt die allgemeine Lautstärke des gesamten Soundprojekts wählbar auf einen Wert zwischen -36dB (3%) bis +6dB (200%). +0dB (100%) entspricht dem Wert =64 in **CV #266**, 200% entspricht dem Wert =128. Es kann auch Mute (0%) ausgewählt werden. Ebenso kann für die Lautstärke auch eine Script-CV ausgewählt werden. Die Fade-Zeit von der aktuellen Lautstärke des Soundprojekts bis zur gewünschten Lautstärke im Script kann in 0,1s-Schritten als Fixwert eingestellt werden (vergleichbar mit **CV #314** [Mute Ein-/Ausblendzeit]), oder als Script-CV ausgewählt werden.

### Abfragen:

**Abfrage Taste:** Prüft, ob eine Funktionstaste **ein-** oder **ausgeschaltet** ist und legt fest, wohin gesprungen werden soll, wenn die Prüfung positiv ausfällt. Durch Klick in das Feld „dann gehe zu“ wird ein State hineingeschrieben und mit ENTER bestätigt. Die gewünschte F-Taste kann auch durch eine der Script-CVs #980-#1011 bestimmt werden; Werte =0 bis =28 entsprechen den Tasten F0 bis F28. Zudem kann die MAN-Taste (nur als Fixwert) abgefragt werden.

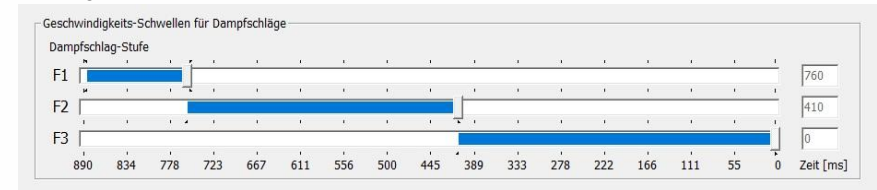
**Abfrage Parameter:** Vergleicht ob der abgefragte **Parameter** (siehe anschließende detaillierte Beschreibung) gleich, ungleich, größer, kleiner, als der definierte Wert (Fixwert oder aus CV) ist. Es kann auch mit einem anderen Parameter verglichen werden, z.B. ob die Fahrgeschwindigkeit gleich der Zielgeschwindigkeit ist. Genau wie bei „Abfrage Taste“ definiert man im letzten einstellbaren Parameter, wohin das Script springen soll, wenn diese Prüfung positiv ausfällt. Wenn diese Prüfung negativ ausfällt, wird der Ablauf im nächsten State fortgeführt.

Abfragen
Abfrage Taste
Abfrage Parameter
Abfrage Reed-Input
Abfrage Event
Abfrage Variable
CV lesen (nur MS)
CV-Bit lesen (nur MS)

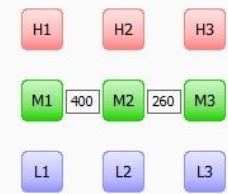
Die Folgenden Parameter können eingestellt oder abgefragt werden:

- **Fahrgeschwindigkeit:** Aktuelle interne Fahrstufe (0-255<sup>1</sup>).
- **Zielgeschwindigkeit:** Am Fahrregler eingestellte Fahrstufe (0-255), auf die beschleunigt oder verzögert wird.
- **Fahrsound:** Prüft, ob Fahrsound aus- oder eingeschaltet ist. Diese Abfrage ist beim Start von Sounds zu bevorzugen, statt der Abfrage nach der Sound-Taste, z.B. F8.
- **Ziel-Set:** Prüft, auf welches Lokset als nächstes gewechselt wird (z.B. bei Set-Wechsel per Taste CV #345). Der Set-Index ist im Script wie folgt: Set 1 = Index 0, Set 2 = Index 1, usw. Wenn in einem Diesel- oder Elektro-Projekt mit Übergangs-Sets gearbeitet wird (z.B. bei Dual-Fahrzeugen, CV #346 Bit2), hat das erste Set den Index =0 und das 4. vollständige Set den Index =1. Die Übergangs-Sets können nicht explizit abgefragt werden und der Index-Wechsel findet jeweils mit dem Ende des Übergangs-Sets statt.
- **Aktuelles Set:** Prüft, welches Set momentan abgespielt wird. Bei Dampfprojekten findet ein Set-Wechsel sofort statt. Es ist also das aktuelle Set und das Ziel-Set dasselbe, weshalb nur eine Abfrage nötig ist.
- **Diesel-Stufe:** Prüft, auf welcher Stufe sich der Diesel-Ablauf-Sound befindet. Der Begriff „Diesel“-Stufe wird im Sinne des Sound-Ablaufs verwendet und kann auch z.B. verschiedene Lüfter-Stufen einer Elektrolok enthalten.
- **Timer-Wert:** Zählt von in „Timer laden“ gesetztem Wert hinunter auf 0. Mit dieser Abfrage ist es möglich, innerhalb des laufenden Timers diverse Aktionen auszulösen.
- **Beschleunigung:** Anzahl an Fahrstufen, um die beschleunigt wird. Wenn das Fahrzeug gleichbleibend fährt oder verzögert, liefert das Script den Wert 0.
- **Verzögerung:** Anzahl an Fahrstufen, um die verzögert wird.
- **Ist-Fahrriechung:** Prüft die aktuelle Fahrriechung (vorwärts, rückwärts).
- **Soll-Fahrriechung:** Prüft, welche Fahrriechung die Lok laut Zentrale haben soll (z.B.: Lok noch im Bremsvorgang in „alter“ Richtung vor dem Richtungswechsel).
- **Consist aktiv:** Prüft (Ja, Nein), ob sich der Decoder momentan im Consist-Betrieb befindet.
- **Consist invers:** Prüft, ob in CV #19, Bit 7 = 1 gesetzt (invertiert die Fahrriechung im Consist-Betrieb) und ob Consist-Betrieb aktiv ist.
- **HLU-Limit:** Aktuelles Limit der HLU-Bremsstrecke. Die Werte über die Script-CVs sind: =0: HLU-Notstop (Bremsen ohne Verzögerung), =2: Halt, =3: UH, 4= U, =5: LU, =6: L, =7: FL, =8: Fahrt.
- **ABC-Limit:** Aktuelles Limit der ABC-Bremsstrecke. Die Werte über die Script-CVs sind: =2: Halt, =5: Langsam, =8: Fahrt.
- **DC-Limit:** Aktuelles Limit der DC-Bremsstrecke. Die Werte über die Script-CVs sind: =2: Halt, =8: Fahrt.
- **Akt. Limit:** Niedrigstes Limit der drei vorigen Parameter. Damit diese nicht alle drei im Script abgefragt werden müssen, testet dieser Parameter auf den niedrigsten gefundenen Wert.

- **Dampfschlag-Intervall** (nur bei MS/FS): Fragt vergleichend die Zeit in Millisekunden zwischen den einzelnen Dampfschlägen ab (näheres dazu in der [ZIMO Sound Creator](#) (ZSP) Anleitung).



- **Dampfschlag/Umdrehung** (nur bei MS/FS): Fragt vergleichend den aktuellen Dampfschlag ab (1-16). Die Anzahl der Dampfschläge wird ausschließlich im ZIMO Sound Creator festgelegt (bei Anz. Schläge, 2-16). Zu beachten ist hier, dass der Index und nicht die Zahl des Dampfschlages gilt, also Dampfschlag 1 =Wert 0, Dampfschlag 2 =1, usw...
- **Dampfschlag-Geschwindigkeitsstufe** (nur bei MS/FS): Fragt vergleichend die Stufe ab, die aktuell im Ablauf abgespielt wird („horizontale“ Dampfschlag-Stufe 1-10).
- **Dampfschlag-Laststufe** (nur bei MS/FS): Fragt vergleichend die aktuelle Laststufe H, M oder L ab.
- **Entwässern aktiv** (nur bei MS/FS): Fragt vergleichend den Zustand der Entwässerungsfunktion ab.
- **Aktueller Betriebsmodus** (nur bei MS/FS/MN): Fragt den aktuellen Modus Digital oder Analog ab. „Keiner“ = Wert 0, „Digitalbetrieb“ =1, Analogbetrieb =2.



**Abfrage Reed-Input:** Der Zustand eines Eingangs (1-4) wird abgefragt. Dieser Zustand kann entweder „Aktiv“ oder „Inaktiv“ sein, beziehungsweise „Event aufgetreten“ oder „kein Event aufgetreten“ liefern. Ändert sich der Zustand eines Reed-Inputs, wird ein „Event“ ausgelöst, bleibt der Zustand unverändert (Ein oder Aus), tritt kein Event auf. Der Eingang kann über eine CV ausgewählt werden, wobei die Werte 0-3 den Eingängen 1-4 zugeordnet sind.

**Abfrage Event:** Fragt vergleichend den Zustand eines Events (1-16) ab. Diese können in Soundsamples gesetzt oder in einem anderen Script ausgelöst werden. Als Vergleichswert gelten in Scripts folgende Werte: 0 = aus/links, 1 = ein/rechts, 2 = Mitte. Die beliebigen höheren Werte bis 255 können ebenso abgefragt werden.

**Abfrage Variable:** Fragt den Wert einer gesetzten Variablen (1-4) ab. Variablen werden zum Zählen bestimmter Aktionen verwendet oder zum „Speichern“ bestimmter Informationen. Als Beispiel: nach jeder dritten Betätigung einer Funktionstaste (wenn ihre Funktion ebenfalls über das Script abgefragt wird) soll ein zusätzlicher Sound ausgelöst werden. Der Variablen wird hierfür der Wert 3 zugewiesen, und nach jeder Betätigung der Taste wird über den Befehl „Variable ändern“ der Wert um -1 verringert. Sobald der Wert 0 erreicht, wird der Sound abgespielt. Es können auch generierte Zufallszahlen, CV-Werte oder CV-Bits in Variablen abgelegt werden.

**CV Lesen** (nur bei MS/FS/MN): Liest im eingestellten Bereich den Wert einer CV aus und setzt diesen in einer Variablen 1-4.

**CV-Bit Lesen** (nur bei MS/FS/MN): Liest ein Bit der eingestellten CV aus und setzt dieses in einer Variablen 1-4.

<sup>1</sup> Je nach eingestelltem Fahrstufenmodus nachzurechnen. Bei 128 Fahrstufen muss der Wert dementsprechend ungefähr mal 2 gerechnet werden

**Setzen:**

**Lok steuern:** Setzt Parameter für die Steuerung der Lok und mittels „Kontrolle“ wird die Art der Beeinflussung definiert. Folgende Steuerbefehle stehen zur Verfügung:

- **Keine Beeinflussung:** wird verwendet, um die folgenden Befehle wieder aufzuheben.
- **Fahrstufe vorgeben:** ungeachtet der aktuell eingestellten Fahrstufe wird hiermit eine bestimmte Fahrstufe vorgegeben. Dies kann auch Fahrstufe 0 sein.
- **Fahrstufe begrenzen:** hiermit wird eine maximale Fahrstufe vorgegeben, die nicht überschritten werden kann.
- **Fahrstufe reduzieren:** reduziert die interne Fahrstufe (von 255) verhältnismäßig zur eingestellten Fahrstufe, z.B.: Wert 128 reduziert etwa auf die Hälfte, Wert 64 etwa auf ein Viertel.
- **Nur Richtung setzen:** erzwingt eine Fahrrichtung (vorwärts, rückwärts, Ost, West, gegen [aktuelle] Fahrrichtung).
- **Notstopp:** hiermit wird ein Notstopp eingeleitet. Die Bremsverzögerung für den Notstopp kann in CV #111 eingestellt werden.
- **Beschleunigung verhindern:** wenn dieser Befehl aktiv ist, kann man nur verzögern oder mit gleicher Geschwindigkeit fahren, aber eben nicht beschleunigen.

Jeder neue Befehl unter **Lok steuern** hebt den vorherigen auf! Das trifft auch für entsprechende Befehle in anderen Scripts zu: wird in einem Script die Fahrstufe auf =0 gesetzt, in einem anderen aber „Keine Beeinflussung“ gewählt, wird nur der letzte Befehl ausgeführt.

In „**Fahrstufe**“ wird die interne Fahrstufe 0-255 definiert. Im letzten Parameter definiert man die Richtung, in der die Beeinflussung gilt (in Fahrrichtung = aktuelle Fahrrichtung, gegen Fahrrichtung = entgegen der aktuellen Fahrrichtung, Vorwärts = Fahrrichtung vorwärts der Lok, Rückwärts = Fahrrichtung rückwärts der Lok, Ost, West). Wenn die Fahrstufe über CV ausgewählt wird, ist eine der Script-CVs #980-#1019 auszuwählen.

**Timer laden:** Hier wird ein Timer eingestellt, der im Hintergrund abläuft. Der Timer zählt je nach gewählter Zeit-Einheit vom eingestellten Wert auf Null zurück. Die Einheiten sind 1ms, 10ms, 100ms, 1s, 10s und 100s bei MS/FS/MN-Decodern. Bei MX-Decodern kann nur 100ms ausgewählt werden. Wenn nach einer bestimmten Zeit eine Aktion ausgelöst werden soll, wird der Parameter „Timer-Wert“ in „Abfrage Parameter“ geprüft! Maximal kann ein Timer bei MS-Decodern auf den Wert 255 mit der Einheit 100s (also 25500 Sekunden, etwa 7 Stunden) gestellt werden. Es können mehrere Timer hintereinander mit unterschiedlichen Einheiten eingerichtet werden. Gleichzeitig funktioniert das aber nicht und es wird jeweils der zuletzt gestartete Timer abgefragt. Der Zeitwert kann auch als Script-CV geschrieben werden.

**Diesel-Stufe fixieren:** Aktiviert eine fixe Dieselstufe (Wert 0= Stand, Wert 20= Dieselstufe F20 [nicht zu verwechseln mit Taste F20]). Damit kann das Script die Kontrolle über den Dieselablauf<sup>2</sup> übernehmen, also eine bestimmte Stufe im Ablauf fixieren ungeachtet des tatsächlichen Ablaufs passend zur aktuellen Geschwindigkeit. Der Decoder läuft alle bis zur ausgewählten Stufe durch und es empfiehlt sich in einer weiteren Abfrage diese „Ziel“-Diesel-Stufe abzufragen, falls diese für den Verlauf des Scripts relevant ist. Deaktiviert man die Fixierung mit „beenden“ in einem der nachfolgenden States, findet der Ablauf selbst den Weg zurück zur passenden Dieselstufe.

Setzen
Lok steuern
Timer laden
Diesel-Stufe fixieren
Ausgang einschalten
Event setzen
FA Effekt setzen
Variable setzen
Variable/Event ändern
Set Parameter
Set Pwm Gruppe
CV schreiben (nur MS)
CV-Bit schreiben (nur MS)

**Ausgang einschalten:** Schaltet einen Funktionsausgang ein (auch über Script-CV zu bestimmen; FA0v =0, FA0r =1, FA1 =2, FA2 =3 usw.), und definiert im folgenden Parameter die Dimmung (den PWM-Wert in %) für diesen Ausgang. Ebenso kann hier der Funktionsausgang auch wieder ausgeschaltet werden und Dimmwerte können sowohl über Script-CV oder eine Variable ermittelt werden. Über das Mapping oder in Samples eingeschaltete Ausgänge können allerdings nicht über ein Script ausgeschaltet werden; diese müssten komplett über das Script verwaltet werden. Die einzige Möglichkeit besteht darin, einen solchen Ausgang einer PWM-Gruppe im Function Mapping zuzuordnen (näheres dazu in den MX- und MS-Decoder Anleitungen Kapitel „Schweizer Mapping“ oder in den Anleitungen zu ZIMO Sound Creator und ZIMO Sound Editor) und diese Gruppe über Set PWM-Gruppe auf 0 zu setzen. Die Einstellung der Effekte wird weiter unten beschrieben.

**Event setzen:** Wenn man diesen Befehl verwendet, sollte man zunächst sichergehen, dass ein Event nicht schon anderweitig in Verwendung ist. Events gelten global im gesamten Projekt, ungeachtet wo sie ausgelöst werden. Für die Samples bietet sich hier ein Blick in die Übersicht. In Scripts kann man Events vorrangig dazu benutzen, mehrere Scripts miteinander kommunizieren zu lassen. Wenn in einem Script eine bestimmte Aktion erreicht ist, kann man ein Event (1-16) mit einem beliebigen Wert 0-255 setzen und dieses in einem anderen Script abfragen.

Möchte man etwas mit einem Sound auslösen, muss man Eventmarker in einem der verwendeten Samples setzen. Dafür im ZIMO Sound Creator im Feld **User Samples** oder dem ZIMO Sound Editor im Fenster **Samples** das Soundfile auswählen und bearbeiten. Im ZIMO Sound Creator kann man das Sample mit Rechtsklick „Loopbereich wählen“ öffnen (näheres dazu in der [Anleitung](#)); im ZIMO Sound Editor ist dies nur per „Position“ möglich, außer bei einem selbst hinzugefügten Sample, wo man die Marker beim Importieren setzen kann.

**FA Effekt setzen:** Damit kann man Effekte auf den ausgewählten Funktionsausgang legen. Dieser Funktionsausgang muss nicht zwingend über ein Script eingeschaltet werden; der Befehl überschreibt den Effekt, der über CV eingestellt wurde.

**Variable setzen:** Hiermit kann eine beliebige Variable 1-4 gesetzt werden und ihr ein Wert 0-255 gegeben werden. Variablen wirken nur in dem Script, in welchem sie gesetzt wurden (anders als Events).

**Variable/Event ändern:** Addiert oder subtrahiert einen Wert von 1 von einer Variablen 1-4, oder einem Event 1-16. Soll mehr als der Wert 1 addiert oder subtrahiert werden, wird dieser Befehl einfach mehrfach hintereinander gesetzt.

**Set Parameter:** Folgende Parameter stehen zur Auswahl (Stand MS-FW 5.21.1 und MX-SW 40.22):

- **SET-Offset:** setzt ein Diesel-Set. Wert 0 = Diesel-Set 1, Wert 1 = Diesel-Set 2... (MS und MX)
- **Dampftakt CV828:** verwendet wahlweise den Dampfschlagtakt von CV #828 (noch nicht implementiert).
- **Zylinderventil offen:** aktiviert oder deaktiviert „Entwässern“ (noch nicht implementiert).
- **Mute aktivieren:** aktiviert oder deaktiviert die Mute-Funktion incl. Fade-Zeit CV #314 (MS und MX).
- **Takt aktivieren:** legt den Impuls des Rauchventilators auf einen Funktionsausgang (noch nicht implementiert).
- **Rangiertaste aktivieren:** aktiviert oder deaktiviert die Beschleunigungsdeaktivierung laut CV #124 (aktuell nur bei MX).
- **Motor invertieren:** invertiert die Richtung des Fahrmotors (nur bei MS/FS/MN).
- **Fahr sound ein/aus:** egal in welchem Fahr-Zustand der Sound sich befindet, wird hiermit der Fahr sound auf AUS oder Ein gezwungen. Man kann diesen Befehl mit „Diesel-Stufe

<sup>2</sup> Dieselablauf: Beim Beschleunigen oder Bremsen durchläuft der Dieselmotor, oder bei E-Loks der Lüfter, verschiedene Stufen, also Stand, F1, F2, etc. und Übergangssamples (F1-F2, F2-F3, bzw. F4-F3, F2-F1, usw.). Siehe dazu die ZSP-Anleitung.



fixieren“ weiter oben vergleichen. Mögliche dazwischen liegende Dieselstufen werden der Reihe nach abgespielt. (nur bei MS/FS).

- **Lichter invertieren:** invertiert die Richtung aller richtungsabhängigen Lichter (nur MS/FS/MN).
- **Lokrichtung invertieren:** invertiert die Rückmeldung der Protokoll-Richtung (nur MS/FS/MN).

**Set PWM-Gruppe:** Hiermit können die Einstellungen der PWM-Gruppen im Function Mapping gesetzt oder geändert werden. Der PWM-(Dimm-)Wert kann in 3%-Schritten eingestellt werden und genau wie im Function Mapping auch kann man ein Blinken einrichten, oder einen Effekt unterdrücken.

**Set CV** (nur bei MS/FS/MS): Setzt eine beliebige CV auf einen festgelegten Wert (die zulässigen CV-Werte sind in der Betriebsanleitung für MS-Decoder angegeben und müssen beachtet werden!). Dieser Wert kann entweder als Konstante gesetzt oder auf eine Variable bzw. ein Event angewendet werden.

**Set CV Bit** (nur bei MS/FS/MN): hier können Bits 0-7 in beliebigen CVs auf 0 oder 1 gesetzt werden.

**Read CV** (nur bei MS/FS/MN): hier kann der Wert einer beliebigen CV ausgelesen werden und anschließend in einer Variablen 1-4 abgelegt werden.

**Read CV Bit** (nur MS/FS/MN): liest Bit 0-7 einer beliebigen CV aus und wendet ihn auf eine Variable 1-4 an.

**Sonstige:**

**Gehe zu:** Setzt lediglich bei Erreichen dieses States einen Pfad zu einem anderen State.

**Zufallszahl generieren:** Generiert einen zufälligen Wert zwischen „Wert von -“ und „bis -“, der entweder im Script fix definiert wird oder über Script-CVs ausgeführt und nachträglich auf dem Decoder geändert werden kann. Die Zufallszahl wird wahlweise auf eine Variable (1-4) oder einen Timer angewandt, welcher gleichzeitig gestartet wird. Die Zeiteinheit kann aus verschiedenen Einheiten gewählt werden. Direkt im Anschluss kann ein Timer-Wert im „Abfrage Parameter“ abgefragt werden.

**I<sup>2</sup>C Command:** ZIMO-interne Testfunktion für MX645 zur Ausgabe von Signalen an I<sup>2</sup>C-Erweiterungen; aktuell aber nicht implementiert.

Sonstige
Gehe zu State
Zufallszahl generieren
I2C Command

**Beispiel:**

Am einfachsten lässt sich die Anwendung von Scripts durch ein Beispiel erklären (hier eines der Muster-Scripts „Pffiff richtungsabhängig“, welche [hier](#) auf der [ZIMO Sound Datenbank](#) heruntergeladen werden können):

Statenname	Statenummer	Kommando	Parameter
->	0	Abfrage Parameter	Fahrsound ist gleich Fixwert aus dann gehe zu State 0
	1	Abfrage Parameter	Fahrgeschwindigkeit ist gleich Fixwert 0 dann gehe zu State 0
	2	Abfrage Taste	Wenn F4 ist Aus dann gehe zu State 0
	3	Abfrage Parameter	Ist-Fahrtrichtung ist gleich Fixwert Rückwärts dann gehe zu State 8
	4	Sound starten	BR57_Pfiff_vor.wav Lautstärke = -3 dB (71%) Sound einmal abspielen Bei Ende alles abspielen Sound sofort starten
	5	Auf Sound Ende warten	
->	6	Abfrage Taste	Wenn F4 ist Ein dann gehe zu State 6
	7	Gehe zu State	0
->	8	Sound starten	BR57_Pfiff_rück.wav Lautstärke = -3 dB (71%) Sound einmal abspielen Bei Ende alles abspielen Sound sofort starten
	9	Auf Sound Ende warten	
->	10	Abfrage Taste	Wenn F4 ist Ein dann gehe zu State 10
	11	Gehe zu State	0
	12	-	-

Wie dieses Script „gelesen“ wird:

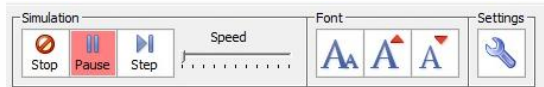
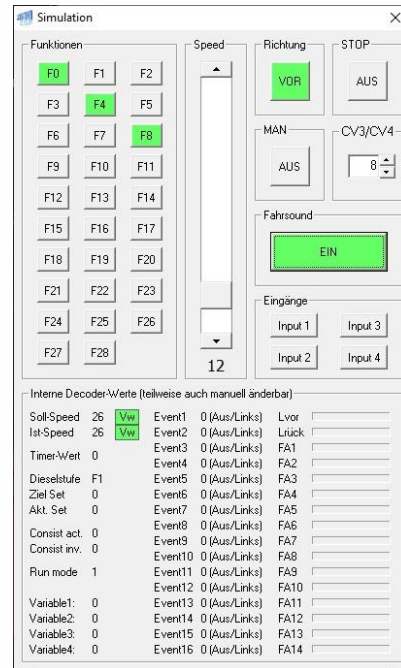
- Ist Fahrsound aus?
  - Ja? → bleibe in State 0
  - Nein? → gehe weiter zu State 1
- Ist Fahrgeschwindigkeit 0?
  - Ja? → gehe zurück zu 0
  - Nein? → gehe weiter zum nächsten State 2
- Ist Funktionstaste F4 aus?
  - Ja? → gehe zurück zu 0
  - Nein? → weiter zu State 3
- Welche Fahrtrichtung ist aktuell eingestellt?
  - Rückwärts? → gehe zu State 8
  - Vorwärts? → gehe weiter zum nächsten State 4
- Startet Sound von Pffiff vorwärts.
- Wartet, bis der Pffiff zu Ende gespielt hat.
- Ist Funktionstaste F4 noch ein?
  - Ja? → bleibe auf State 6
  - Nein? → gehe weiter zum nächsten State, und anschließend zu 0.

Der Abfragen-Ablauf von States 4-7 wiederholt sich in den States 8-11 für den Pffiff für Fahrtrichtung rückwärts. Die Abfrage nach dem Fahrsound könnte auch jeweils direkt vor einem „Sound starten“ durchgeführt werden.

## Scripts simulieren:

Sobald das Script fertig programmiert ist, kann es sowohl im ZIMO Sound Creator als auch im ZIMO Sound Editor auf Fehler getestet werden. Auch zwischendurch besteht die Möglichkeit, das Script zu testen und zu überprüfen, indem es in einer Simulation ausgeführt wird. Dies erfolgt durch den Knopf „**Simulation Run/Stop**“ unten links im Script-Editor.

Die Simulation funktioniert derzeit (ZSP V1.26.00) nicht mit allen Abläufen und Parametern, da ZIMO Sound Creator und ZIMO Sound Editor in einigen Fällen nicht in der Lage sind, bestimmte Funktionen nachzubilden (wie Servo-Bewegungen, Abfragen nach Diesel-Set, etc.). Grundsätzlich lässt sich aber jedes Script durchlaufen, um zu überprüfen, ob es an einer Stelle hängen bleibt oder in einem Loop durchläuft. Wenn das Script zu schnell ausgeführt wird und die einzelnen Schritte nicht nachvollzogen werden können oder ein Fehler auftritt, dessen Ursache nicht genau ermittelt werden kann, besteht die Möglichkeit, das Script Schritt für Schritt ablaufen zu lassen. Dafür wird auf „Pause“ geklickt, um die Ausführung anzuhalten. Anschließend kann mit „Step“ jeder Schritt im Script einzeln durchgegangen werden. Die Geschwindigkeit des Durchlaufs lässt sich außerdem über die Option „Speed“ anpassen.



Das Beschleunigungs- und Verzögerungsverhalten kann im Feld CV3/CV4 eingestellt werden. Wenn in einem Script der Fahrsound geprüft wird, kann in der Simulation „Fahrsound EIN“, oder die entsprechende Taste aktiv sein. Es gibt die Möglichkeit Input 1-4 zu simulieren und es werden Variablen, Events und Funktionsausgänge angezeigt, die in diesem Script ausgelöst werden. Bei Ziel Set und Akt. Set kann man in das Feld klicken und die gewünschten Werte eintragen.

Es wird immer nur das gerade geöffnete Script getestet – mögliche Interaktionen mit anderen Scripts können nicht getestet werden.

## Nächste Schritte, sobald das Script erfolgreich funktioniert:

Mit einem Klick auf „OK“ rechts unten wird das Script gespeichert und zum Hauptfenster des Scripts zurückgekehrt.

Bei „Cancel“ wird der Script-Editor geschlossen und nicht gespeichert!

## Wenn das Script NICHT wie gewünscht funktioniert:

Die Arbeit mit den Scripts zeigt schnell, dass diese sehr umfangreich und komplex werden können. Ein nützliches Hilfsmittel für die Fehleranalyse ist der Simulator. Falls der Fehler auf diese Weise nicht identifiziert werden kann, sollte das Script gespeichert, ein neues Script angelegt und der fehlerhafte Teil mit möglichst wenigen States nachgestellt werden. Es kann auch hilfreich sein, das gewünschte Verhalten, das durch das Script ausgelöst werden soll, schriftlich festzuhalten.





ZIMO Elektronik GmbH

ZIMO ELEKTRONIK GmbH, Schönbrunner Straße 188, 1120 Wien, Österreich | [www.zimo.at](http://www.zimo.at) | Änderungen und Irrtümer vorbehalten.

RailCom ist ein Markenzeichen der Lenz GmbH, mfx ist ein Markenzeichen der Märklin & Cie GmbH